

ІНФОРМАЦІЙНА СИСТЕМА ДОСЛІДЖЕННЯ ЕКОЛОГІЧНОГО СТАНУ ПРИРОДНО-ЗАПОВІДНИХ ТЕРИТОРІЙ

*Леся Угрин, Senior Lecturer, Мар'яна Баран, PhD, As. Prof,
Національний університет «Львівська політехніка», Україна,
Тоніс Теппад, викладач аналітичного вибору будівельних конструкцій, науковий співробітник;
Інститут лісового господарства та сільського господарства, Естонський університет природничих наук,
Естонія,
e-mail: Lesya.e.uhryn@lpnu.ua;
e-mail: tonis.teppand@emu.ee*

Анотація

Стаття висвітлює питання моніторингу та дослідження екологічного стану територій, що мають статус природно-заповідних. У роботі описано створення інформаційної системи з елементами штучного інтелекту для вивчення екологічного стану природно-заповідних територій. В результаті аналізу даних моніторингу визначено первинні дані для роботи та сформовано базу даних. Особливу увагу приділено опису інформаційної моделі об'єкта та обраних показників для його адекватного функціонування. Вбудована система штучного інтелекту буде навчена на основі сформованої моделі для підвищення швидкості та якості визначення стану території заповідника. Велика увага буде приділена можливості швидкого та безперебійного отримання даних з датчиків. Оптимізований за шаблоном код для відправки даних з датчиків легко допоможе швидко передавати великі обсяги даних, а сформована динамічна модель інтерфейсу користувача дозволить адміністраторам збільшувати можливості системи в залежності від своїх потреб. Проаналізовано обрані технології створення інформаційної системи, мікросервіси, платформи потокової передачі подій Apache Kafka та Confluent Platform. Результати дослідження представлені у вигляді розробленої та апробованої інформаційної системи.

Ключові слова

Інформаційна система, штучний інтелект, мікросервіси, природно-заповідні території.

1. Вступ

Дослідження та моніторинг екологічного стану природно-заповідних територій України мають першочергове значення в сучасному екологічному ландшафті. Актуальність цієї теми підкреслюється безперервною деградацією природних екосистем, що посилюється такими факторами, як промислова діяльність, зміна клімату та, зокрема, триваючий конфлікт у регіоні. Оскільки цей тиск посилюється, потреба у всебічному розумінні екологічної динаміки в природних заповідниках стає все більш критичною. У контексті України, де збереження різноманітної флори та фауни є невід'ємною частиною ідентичності нації та екологічної стійкості, ця тема стає однією з пріоритетних. Екологічні виклики, спровоковані конфліктом, включаючи ракетні удари по ключових інфраструктурах і викид небезпечних речовин, посилюють вразливість природних заповідників. Це підкреслює нагальну потребу в ретельному дослідженні та постійному моніторингу для оцінки впливу на біорізноманіття, якість повітря та загальний екологічний баланс. Інформаційні системи зі штучним інтелектом ефективно збирають, зберігають, систематизують і обробляють великі обсяги інформації та отримують результати, отримані при дослідженні природно-заповідних територій.

2. Проблеми і труднощі

Актуальність цього дослідження ще більше посилюється на тлі триваючого конфлікту в Україні, обставини, які загострили екологічні проблеми, з якими стикається нація. Ракетні удари по життєвоважливих об'єктах інфраструктури, включаючи нафтохранилища та електростанції, призвели до миттєвих наслідків для навколишнього середовища та створили довгострокову загрозу через викид різноманітних небезпечних речовин у повітря. Отже, це суттєво перешкоджає ефективному виконанню дослідницької та моніторингової діяльності, посилюючи потребу в надійній та адаптивній системі моніторингу. Доступ до датчиків, з яких збираються дані та передаються в систему моніторингу, стає проблемою. Збір даних з деяких областей стає обмеженим, а іноді й недоступним.

3. Мета

Метою роботи була розробка інформаційної системи з навченою нейронною мережею для обробки даних та оцінки якості території, яка може бути інтегрована в різноманітні середовища, пріоритетними є природно-

заповідні території. Інформаційна система доповнена програмами для швидкої відправки даних із датчиків у робоче середовище, а динамічний інтерфейс користувача дозволить системному адміністратору легко додавати нові показники для дослідження.

4. Дослідження та розробка інформаційної системи

Глобальна дискусія про збереження навколишнього середовища та стала практикою ще більше посилює актуальність цієї теми. Дослідження екологічного стану природно-заповідних територій в Україні виходить за рамки регіональних проблем, охоплюючи ширші екологічні, культурні та глобальні виміри. Ця робота спрямована на критичну потребу у всебічному розумінні поточного екологічного стану та закладає основу для інформованих методів збереження та сталої екологічної політики. Основною темою цього дослідження є підкреслення незамінної ролі використання інформаційних систем у дослідженні та моніторингу екологічного стану на територіях природно-заповідного фонду та його вирішального внеску в поточну природоохоронну роботу, спрямовану на збереження багатого біорізноманіття України.

Безпечне середовище не тільки відповідає законодавчим критеріям, але й відповідає встановленим стандартам і нормам, охоплюючи різні аспекти, такі як чистота, стійкість ресурсів, екологічна стабільність, санітарні умови, різноманіття видів і здатність задовольняти інтереси громадян. Основний напрямок цього дослідження зосереджено навколо всебічної оцінки екологічного стану природно-заповідних територій України, керуючись головною метою створення доступної та зручної системи моніторингу. Ця система ретельно розроблена для поширення інформації серед громадськості через онлайн-платформу, таким чином сприяючи більшій екологічній обізнаності.

Після проведення ретельного аналізу відомих систем були виділені переваги та недоліки з конкретними функціями, стратегічно обраними для інтеграції в розроблену інформаційну систему.

Станом на 1 січня 2022 року (початок повномасштабної війни) в Україні діяло 4 біосферні заповідники та 16 державних природних заповідників. Фонд охорони природи забезпечує збереження земель і вод, природних комплексів, об'єктів, підтримує загальну екологічну рівновагу, здійснює фоновий моніторинг навколишнього природного середовища. В Україні заповідні фонди охороняються як державна власність, застосовуються особливі режими охорони, відтворення та використання. Україна вважає цей Фонд невід'ємною частиною світової системи природних територій та об'єктів, що перебувають під особливою охороною. До складу природно-заповідного фонду України входять природні території та об'єкти - природні заповідники, біосферні заповідники, національні природні парки, місцеві ландшафтні парки, заповідні території, пам'ятки природи, заказники, техногенні об'єкти - ботанічні сади, сади, зоопарки та інші парки - пам'ятки садово-паркового мистецтва. У зв'язку з цим Законом України фонди заповідників охороняються як державне майно, встановлюються спеціальні режими їх охорони, відновлення та використання. Україна вважає цей фонд невід'ємною частиною світової системи природних територій та об'єктів, що перебувають під особливою охороною. Заповідники — природоохоронні та науково-дослідні установи загальнодержавного значення, завданням яких є збереження типових або виняткових природних комплексів з усіма їх компонентами в природному стані та збереження автентичності, вивчення процесів і явищ, розробка наукових основ охорона природи. Передбачається охорона навколишнього середовища, ефективне використання природних ресурсів, екологічна безпека. Землі та води, що належать до природно-заповідного фонду, вилучаються з господарського використання. Заповідник є найвищим заказником і природною лабораторією, де проводяться комплексні наукові дослідження. Кожен великий природний комплекс має природоохоронні території. Станом на 1 січня 2025 року через війну частка природно-заповідного фонду значно зменшилася. Близько 20% заповідних територій знаходяться на окупованій території, що створює труднощі в отриманні даних моніторингу та потребує додаткових зусиль для їх збереження та відновлення[1-2].

На основі цих даних та даних проаналізованих існуючих систем ми розробимо сучасну, просту у використанні систему дослідження та моніторингу природно-заповідного фонду України. Система моніторингу перевірятиме дані за трьома напрямками. Зокрема, для оцінки стану повітря будуть використовувати показники дрібного пилу. Це тверді частинки різного діаметру, це можуть бути вихлопні гази, сажа, пил, бруд та інші. PM 2,5 — це тверді частинки діаметром 2,5 мікрона або менше. PM 10 — це тверді частинки діаметром від 2,5 мікрон до 10 мікрон. Для оцінки води буде використовуватися показник Ph води. Також буде індикатор температури [3-4]. Тому розроблена система, основними функціями якої будуть моніторинг та контроль заповідних територій. Система розділена на кілька частин.

Перша частина — програмне забезпечення для швидкої передачі даних у середовище зберігання та передачі даних користувачеві.

Друга частина — це програмне забезпечення для використання цих даних системою та передачі їх для безпосередньої обробки та оцінки штучним інтелектом на основі попередньо підготовленої моделі нечіткої логіки.

Третя частина – система штучного інтелекту, навчена на основі сформованої моделі визначати стан території, що охороняється.

Четверта частина — це сервісне програмне забезпечення, яке передаватиме ці дані в інтерфейс користувача.

П'ята частина – це інтерфейс користувача, за допомогою якого користувач зможе перевіряти та визначати стан навколишнього природного середовища заповідника.

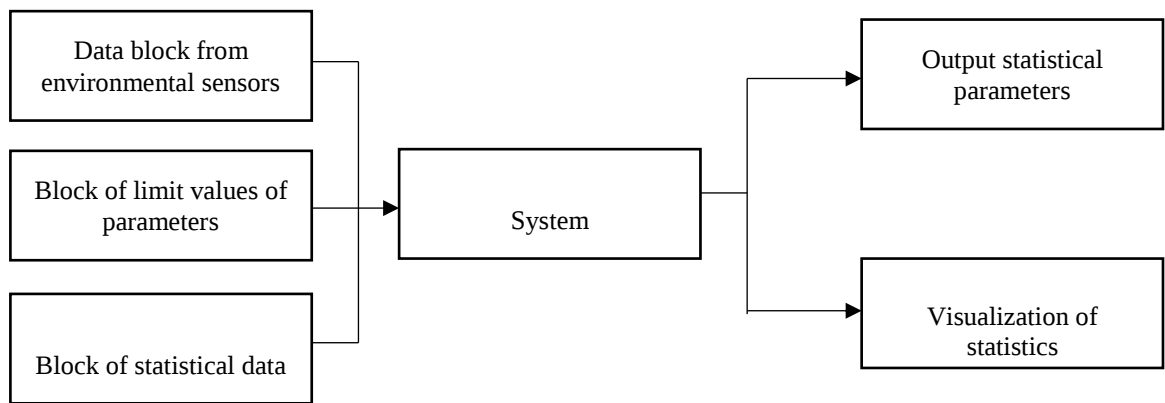


Рис. 1. Інформаційна модель розробленої системи

Використовується інформаційна модель, яка описує параметри, значення та зв'язки об'єктів (рис.1).

Ця діаграма відображатиме ланцюжок обробки даних: дані з датчиків навколишнього середовища надходять у систему та визначаються граничні значення параметрів (мінімальна/максимальна температура, вологість). Після цього дані аналізуються і система об'єднує всі блоки. Результати представлені у вигляді статистичних параметрів і візуалізуються для подальшого аналізу.

В ході роботи була розроблена діаграма варіантів використання UML, написані специфікації акторів і специфікації варіантів використання. UML — це стандартизована мова моделювання, яка використовується в розробці програмного забезпечення для візуалізації, проектування та документування систем (табл. 1-2).

Табл. 1. Специфікація актора

Актор	опис
Admin	Admin має розширений доступ до роботи з системою. Вміння працювати з датчиками та системою прийому та передачі даних. Відповідає за завантаження даних у систему. Відповідає за навчання нейронної мережі для отримання результату дослідження природно-заповідної території України
Користувач	Звичайні користувачі системи мають можливість відкрити систему, знайти дані про заповідник за певний період і самостійно порівняти ці дані. Штучний інтелект проаналізує ці дані та поверне користувачеві реальний результат.

Табл. 2. Специфікації варіантів використання

Варіант використання	опис
Управління системою передачі даних	Адміністратори охоронних територій мають доступ до адмінпанелі, середовища передачі даних і самих датчиків, які будуть завантажувати дані в систему.
Робота з інформацією	Адміністратори заповідної території зможуть підтримувати стан штучного інтелекту (навчена нейронна мережа), який відповідатиме за перевірку стану природної заповідної території на основі даних, отриманих із середовища.
Перегляд інформації	Перегляд інформації буде доступний як користувачу, так і адміністраторам охоронних територій. Будь-який користувач, який бажає переглянути дані, зможе це зробити.

Для розробки такої універсальної системи потрібен значний перелік технологій, щоб зробити її максимально універсальною та легкою для інтеграції в майбутньому. Етапи та інструменти розробки можна розділити на кілька рівнів: розробка інтерфейсу користувача, розробка backend системи, підготовка та розміщення середовища передачі даних, споживання та обробка даних[5].

Були обрані наступні технології: Java як основна мова програмування [6-8]. Apache Kafka — це платформа потокової передачі подій, яка використовується для збору, обробки, зберігання та інтеграції даних (рис. 2).

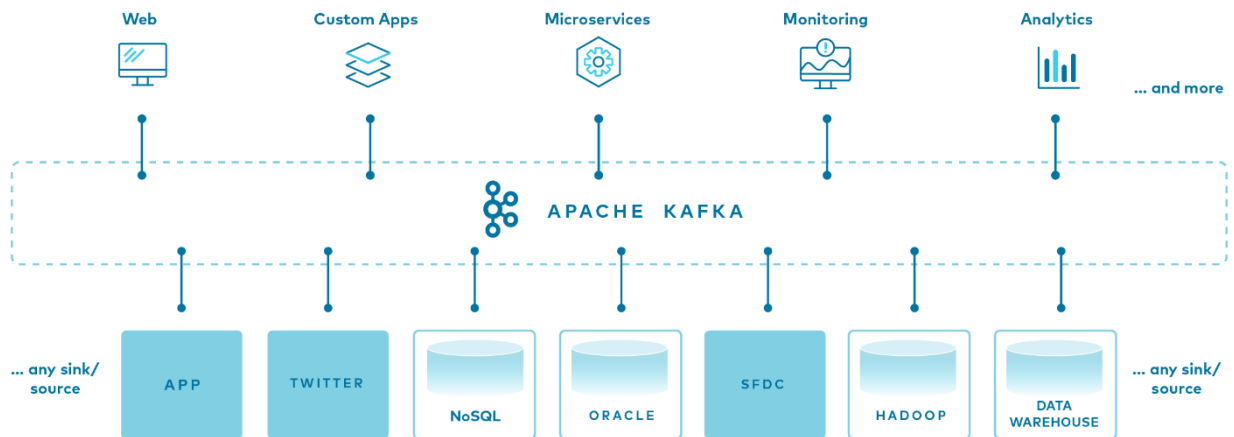


Рис. 2. Модель потоків Кафки

Він має численні варіанти використання, включаючи розподілене потокове передавання, обробку потоків, інтеграцію даних і обмін повідомленнями pub/sub. Платформа Confluent — це повномасштабна потокова платформа, яка дозволяє легко отримувати доступ, зберігати та керувати даними як безперервними потоками в реальному часі.

Confluent розширює переваги Kafka за допомогою функцій корпоративного рівня, одночасно усуваючи складності керування та моніторингу Kafka. Сьогодні понад 80% зі списку Fortune 100 працюють на основі технології потокового передавання, і більшість із них використовує Confluent. Мікросервіси — це популярний архітектурний стиль для побудови систем, які є стійкими, високомасштабованими, незалежно розгорнутими та здатними до швидкого зростання. Але успішні архітектури мікросервісів вимагають іншого підходу до проектування та створення програм (рис.3).

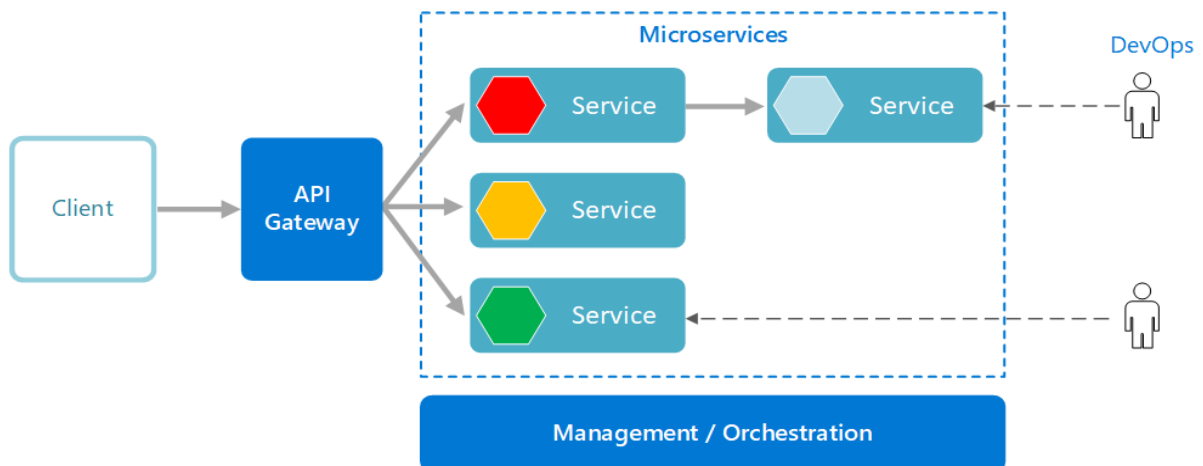


Рис. 3. Дизайн архітектури мікросервісів

Архітектура мікросервісів складається з набору невеликих автономних сервісів. Кожна служба є самодостатньою та має реалізувати одну бізнес-можливість у обмеженому контексті. Обмежений контекст є природним поділом у бізнесі та забезпечує чіткі межі, у межах яких існує модель домену. Мікросервіси невеликі, незалежні та слабко пов'язані. Одна невелика команда розробників може написати та підтримувати службу. Кожна служба є окремою кодовою базою, якою може керувати невелика команда розробників. Послуги можна розгортати незалежно. Команда може оновити наявну службу без перебудови та повторного розгортання всієї програми. Служби відповідають за підтримку власних даних або зовнішнього стану. Це відрізняється від традиційної моделі де окремий шар даних обробляє збереження даних. Служби спілкуються одна з одною за допомогою чітко визначених API. Внутрішня реалізація кожної служби прихована від інших служб. Підтримує потокове програмування, і службам не потрібно спільно використовувати той самий технологічний стек, бібліотеки чи фреймворки.

Написано програмний код для надсилання даних, які використовуватимуться на датчиках. Цей код можна легко використовувати для будь-якого датчика (рис.4).

```

public class NatureProducer
{
    public static Properties loadConfig(final String configFile) throws IOException
    {
        if (!Files.exists(Paths.get(configFile)))
        {
            throw new IOException(configFile + " not found.");
        }
        final Properties cfg = new Properties();
        try (InputStream inputStream = new FileInputStream(configFile))
        {
            cfg.load(inputStream);
        }
        return cfg;
    }

    public static void sendData(String topic, String value) throws IOException
    {
        final Properties props = loadConfig("client.properties");
        props.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG, StringSerializer.class);
        props.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG, StringSerializer.class);
        final Producer<String, String> producer = new KafkaProducer<>(props);
        producer.send(new ProducerRecord<>(topic, UUID.randomUUID().toString(), value));
        producer.close();
    }
}

```

Рис. 4. Код передачі даних датчика

Для передачі даних підготовлено середовище на платформі Confluent. Для цього було розгорнуто кластер Kafka, в якому будуть створені теми або розділи для наших даних (рис.5).

Сервісні облікові записи були створені для організації доступу для запису та читання даних з топіків у цьому кластері, а також для створення ключів (паролів) і топіків для отримання даних від датчиків. Для кожного датчика був виділений окремий топік для збереження цілісності даних при їх зчитуванні із середовища.

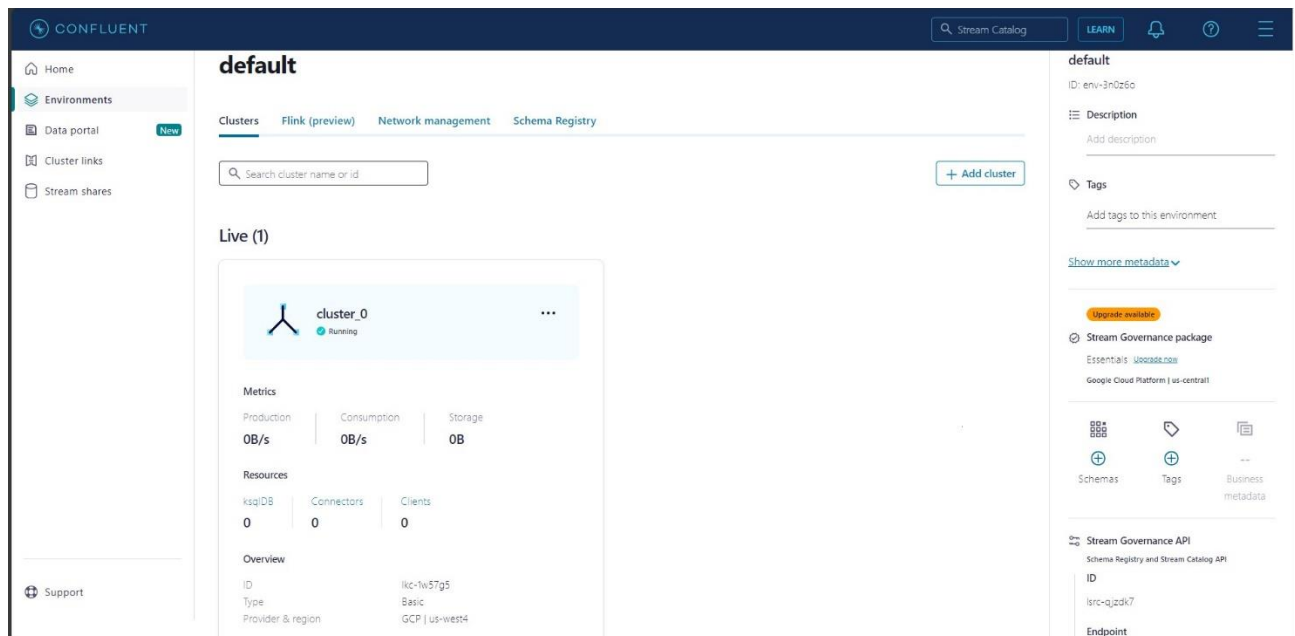


Рис. 5. Кластер Kafka на платформі Confluent

Для розробки інтерфейсу користувача були обрані такі технології: HTML, CSS, JavaScript

Для наповнення бази даних використовувалися дані наявних датчиків моніторингу якості повітря та води, інші дані були отримані з інших доступних та перевірених ресурсів[9]. База даних інформаційної системи спочатку складається з семи таблиць: quality_level, nature_territory, sensors, carbon_monoxide_level_sensor, water_pollution_sensor, air_qualityindex_sensor, water_smell_sensor.

quality_level: ця таблиця міститиме рівень якості для певного вимірюваного параметра (наприклад, рівень якості повітря, рівень якості води тощо), пов'язаний з іншими таблицями.

nature_territory: ця таблиця містить інформацію про природні території, де розташовані датчики, пов'язану з таблицею датчиків.

sensors: основна таблиця, що містить загальну інформацію про всі датчики. Містить такі дані, як унікальні ідентифікатори, розташування, типи датчиків та іншу детальну інформацію.

carbon_monoxide_level_sensor: таблиця, що містить конкретні дані для датчиків рівня чадного газу, пов'язана з таблицею датчиків.

water_pollution_sensor: таблиця з конкретними даними для датчиків забруднення води, пов'язана з таблицею датчиків.

air_qualityindex_sensor: таблиця з конкретними даними для датчиків, які вимірюють загальний індекс якості повітря, пов'язана з таблицею датчиків.

```
public class NatureConsumer
{
    public static void consumeData(List<String> topics) throws IOException
    {
        final Properties props = loadConfig( configFile, "client.properties");
        props.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
        props.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
        props.put(ConsumerConfig.GROUP_ID_CONFIG, "kafka-java-getting-started");
        props.put(ConsumerConfig.AUTO_OFFSET_RESET_CONFIG, "earliest");

        KafkaConsumer<String, String> consumer = new KafkaConsumer<>(props);
        consumer.subscribe(topics);
        while (true)
        {
            ConsumerRecords<String, String> records = consumer.poll(Duration.ofMillis(100));
            for (ConsumerRecord<String, String> consumerRecord : records)
            {
                loadRecordToDb(consumerRecord);
            }
        }
    }
}
```

Рис. 6. Читання даних і завантаження у базу даних

water_smell_sensor: таблиця з конкретними даними для датчиків, які контролюють запах води, пов'язана з таблицею датчиків. Взаємодія між цими таблицями забезпечується ключами, які посилаються на унікальні ідентифікатори датчиків та інші зв'язки. Наприклад, таблиці carbon_monoxide_level_sensor, water_pollution_sensor, air_qualityindex_sensor і water_smell_sensor можуть мати зовнішні ключі, які посилаються на ідентифікатори в таблиці датчиків, що вказує, який датчик виконує вимірювання.

Спроектowana база даних спрощує роботу адміністратора та забезпечує можливість швидкого перегляду необхідної інформації. Зокрема, система передбачає, що адміністратор зможе легко додавати інформацію про показники з датчиків до бази даних за допомогою нашого передавача. Для зчитування даних з тем створено споживач, який забезпечує миттєве зчитування даних з топиків і постійно контролює надходження даних в них(рис.6). Крім того, під час читання даних нашою системою та запису їх у базу даних буде великий набір журналів. В ході роботи було визначено, що дані, які доступні звичайним користувачам про джерела, з яких ми хочемо отримувати інформацію, будуть вноситися в базу адміністратором. Таким чином, система адаптивна і може наповнюватися даними з перевірених джерел для будь-яких регіонів. Домашня сторінка для користувачів містить заголовок із назвою програми та міні-меню (рис.7).

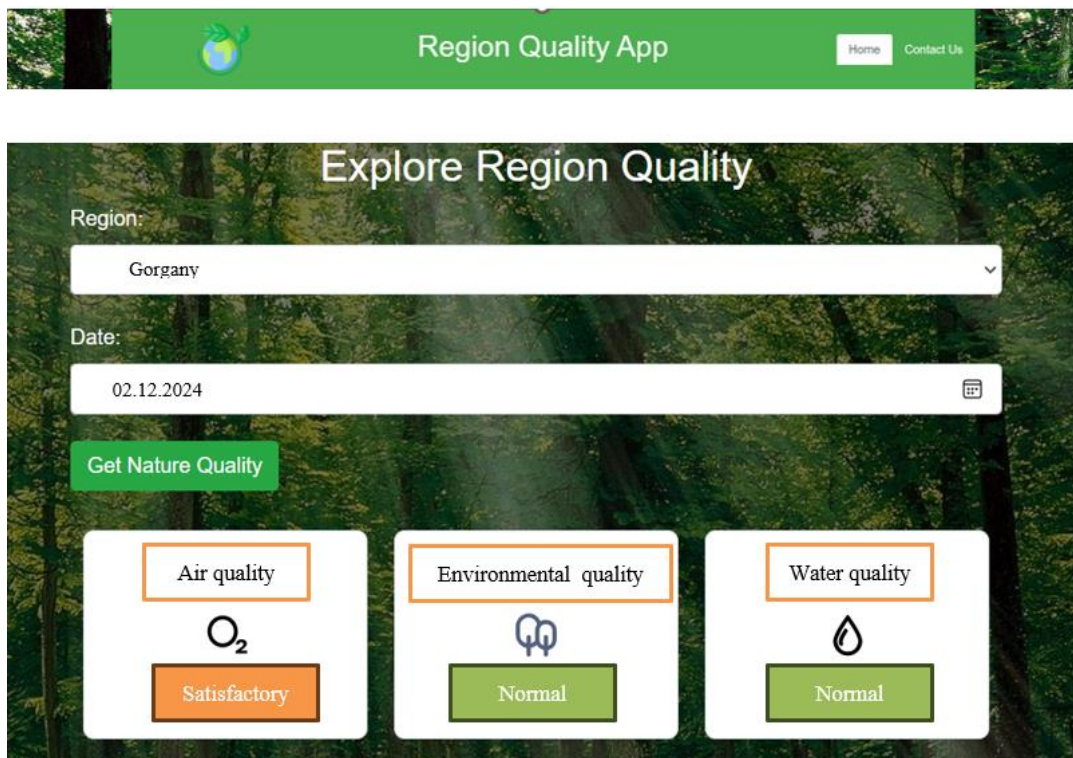


Рис. 7. Результат системи моніторингу

Система розроблена таким чином, що дані з датчиків будуть потрапляти в топіки, система буде використовувати ці топіки для зчитування даних і завантаження їх в базу даних.

```
private static final int NUM_INPUTS = 5; // pm1, pm25, pm10, temperature, water
private static final int NUM_HIDDEN_NODES = 20;
private static final int NUM_OUTPUTS = 5; // Good, Moderate, Harmful, Very Harmful, Dangerous
private static final int LEARNING_RATE_SCHEDULE_EPOCHS = 1000;
private static final double INITIAL_LEARNING_RATE = 0.01;

private static MultiLayerConfiguration.Builder buildNeuralNetworkConfiguration()
{
    return new NeuralNetConfiguration.Builder()
        .seed(123)
        .updater(Adam.builder().learningRate(INITIAL_LEARNING_RATE).build())
        .weightInit(WeightInit.XAVIER)
        .gradientNormalization(GradientNormalization.ClipElementWiseAbsoluteValue)
        .gradientNormalizationThreshold(1.0)
        .list()
        .layer(0, new DenseLayer.Builder()
            .nIn(NUM_INPUTS)
            .nOut(NUM_HIDDEN_NODES)
            .activation(Activation.RELU)
            .build())
        .layer(1, new OutputLayer.Builder(LossFunctions.LossFunction.MCXENT)
            .nIn(NUM_HIDDEN_NODES)
            .nOut(NUM_OUTPUTS)
            .activation(Activation.SOFTMAX)
            .build())
        .backpropType(BackpropType.Standard);
}
```

Рис.8. Конфігурація нейронної мережі

Ця система дозволяє, у разі збою бази даних, повторно отримати всі дані з наших датчиків, зчитавши їх із цих топіків, оскільки платформа дозволяє нам зберігати дані там тривалий час. Для зчитування даних з тем створено кон'юсмер, який забезпечує миттєве зчитування даних і постійно контролює потік даних у них. Для оцінки стану заповідника було навчено нейронну мережу за 5 параметрами, такими як індекс води, температура та три показники повітря (рис.8). Зокрема, цю нейронну мережу навчали на підготовлених вихідних даних із відкритого доступу.

5. Висновки

Таким чином, розроблено цілком актуальну і водночас унікальну систему дослідження та моніторингу природно-заповідного фонду України. На основі аналогових систем виявлено переваги та недоліки існуючих систем, які зіграли значну роль у розробці нашої системи. Розроблена система розділена на три частини: інтерфейс користувача, середовище передачі даних і нейронно-серверну частину системи. Серверна частина

відповідає за цілісність і обробку даних, зокрема, тут адміністратор має можливість додавати нові датчики, оновлювати та видаляти дані. Розроблена система моніторингу є адаптивною і за допомогою цієї системи користувачі матимуть можливість швидко та безперебійно отримувати дані, оскільки оптимізований шаблонний код для надсилання даних із датчиків легко допоможе швидко передавати великі обсяги даних щосекунди, а сформована динамічна модель інтерфейсу користувача дозволить адміністраторам легко розширити можливості системи та адаптувати її для різних територій і регіонів.

Конфлікт інтересів

Автори заявляють, що фінансових чи інших потенційних конфліктів щодо цієї роботи немає.

Список літератури

- [1] Eco Rayon, 2024 [Online]. Available: <https://eco.rayon.in.ua/news/578824-plan-zi-stvorennya-zapovidnikh-teritoriy-v-ukraini-vikonali-lishe-na-35>
- [2] Wow Nature, 2024 [Online]. Available: <https://wownature.in.ua/en/>
- [3] Угрин Л. Є. Математичне моделювання оцінки стану територій природно-заповідного фонду загальнодержавного значення// Інформаційні та інноваційні технології 21 століття: проблеми та перспективи розвитку: матеріали VI міжнародної науково-практичної конференції (м. Катовіце). , 19-20 вересня 2023 р.).
- [4] Угрин Л. Є. Моніторинг екологічної безпеки регіонів природно-заповідного фонду // Інноваційні технології в розвитку сучасного суспільства: тези доповідей VI Міжнародної науково-практичної конференції 10–11 жовтня 2024 р.
- [5] Backend Development: Skills, Salary, Roles & More, 2023 [Online]. Available: <https://www.simplilearn.com/tutorials/programming-tutorial/what-is-backend-development>
- [6] Flanagan, David. "JavaScript: The Definitive Guide." O'Reilly Media, 2020 – 706pp.
- [7] How Java Spring MVC Works: Spring MVC Request, 2023 [Online]. Available: <https://tutorialspedia.com/how-java-spring-mvc-works-spring-mvc-request-flow-explained-step-by-step/>
- [8] Palmer, Boris. "Programming TypeScript: Making Your JavaScript Applications Scale." O'Reilly Media, 2019 - 324pp.
- [9] Beaulieu, Alan. "Learning SQL: Generate, Manipulate, and Retrieve Data." O'Reilly Media, 2020 – 38pp.